

КРАЖЕВСКИЙ АЛЕКСЕЙ

ВАРИАНТ 34

ЛАБОРАТОРНАЯ РАБОТА 8

1.1 Рисование (приложение на языке objective-C)

ЗАДАЧИ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Разобрать пример рисования на языке Objective-C

Входные данные

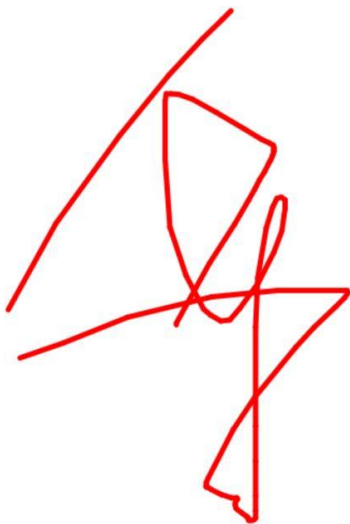
Цвет, размер и форма кисти, используемой для рисования. 4

Выходные данные

Рисунок, состоящий из линий произвольной формы.

Результат выполнения работы:

1:16



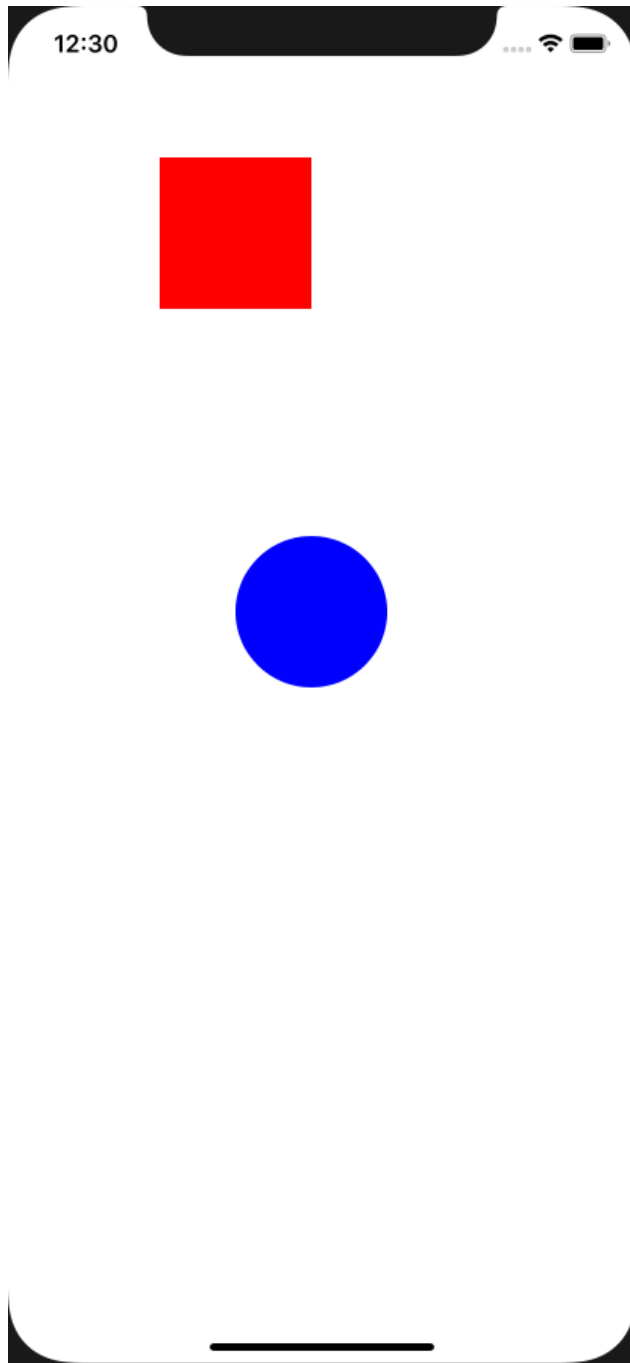
1.2. Создание iOS приложения с простыми графическими объектами на языке Swift

Цель: изучить примеры рисования графических объектов и получить навыки разработки приложений

Результат выполнения:

(реализован пример из первой ссылки(видео) по рисованию квадрата и круга)

Все остальные статьи были изучены.



1.3. Создание iOS приложения на языке Swift с использованием анимации

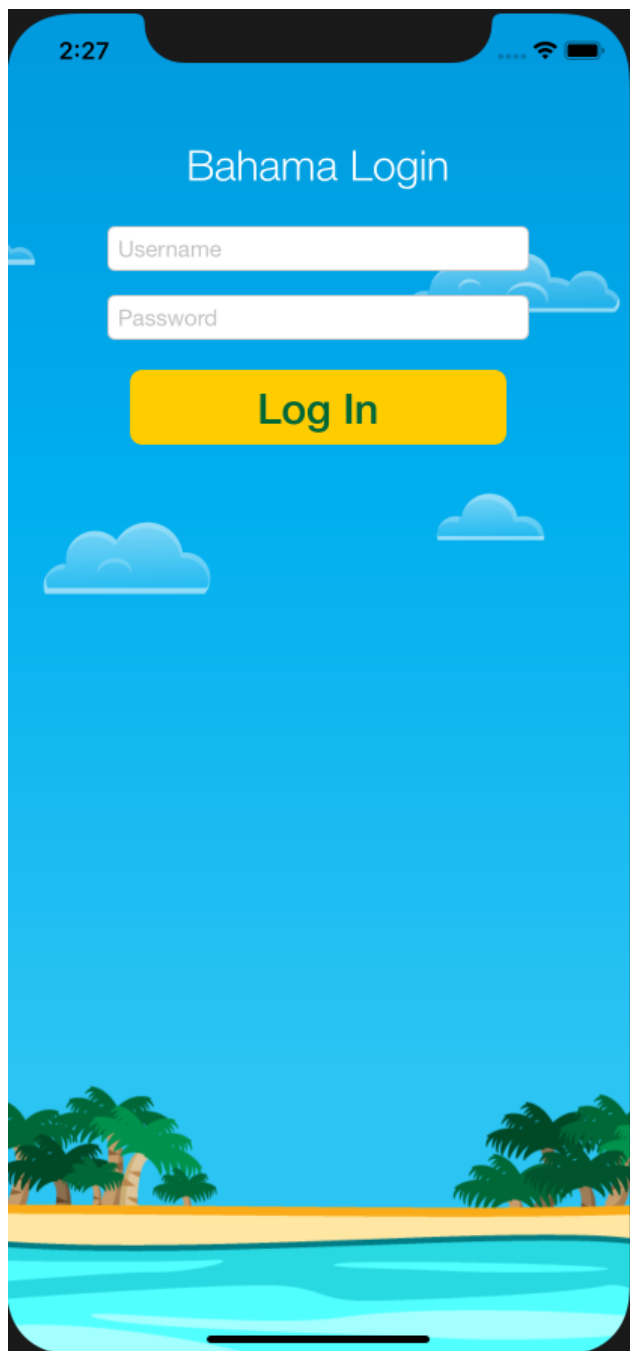
Цель: изучить примеры рисования графических объектов и получить навыки разработки приложений.

Выполнял пример по первой ссылке, скачал прилагающиеся материалы.

Главное окно скачанного проекта:

В ходе выполнения задания была добавлена анимация появления двух полей и анимация движения облачков на фоне.

Все остальные статьи были изучены.



1.4. Создание iOS приложения на языке Swift с использованием жестов

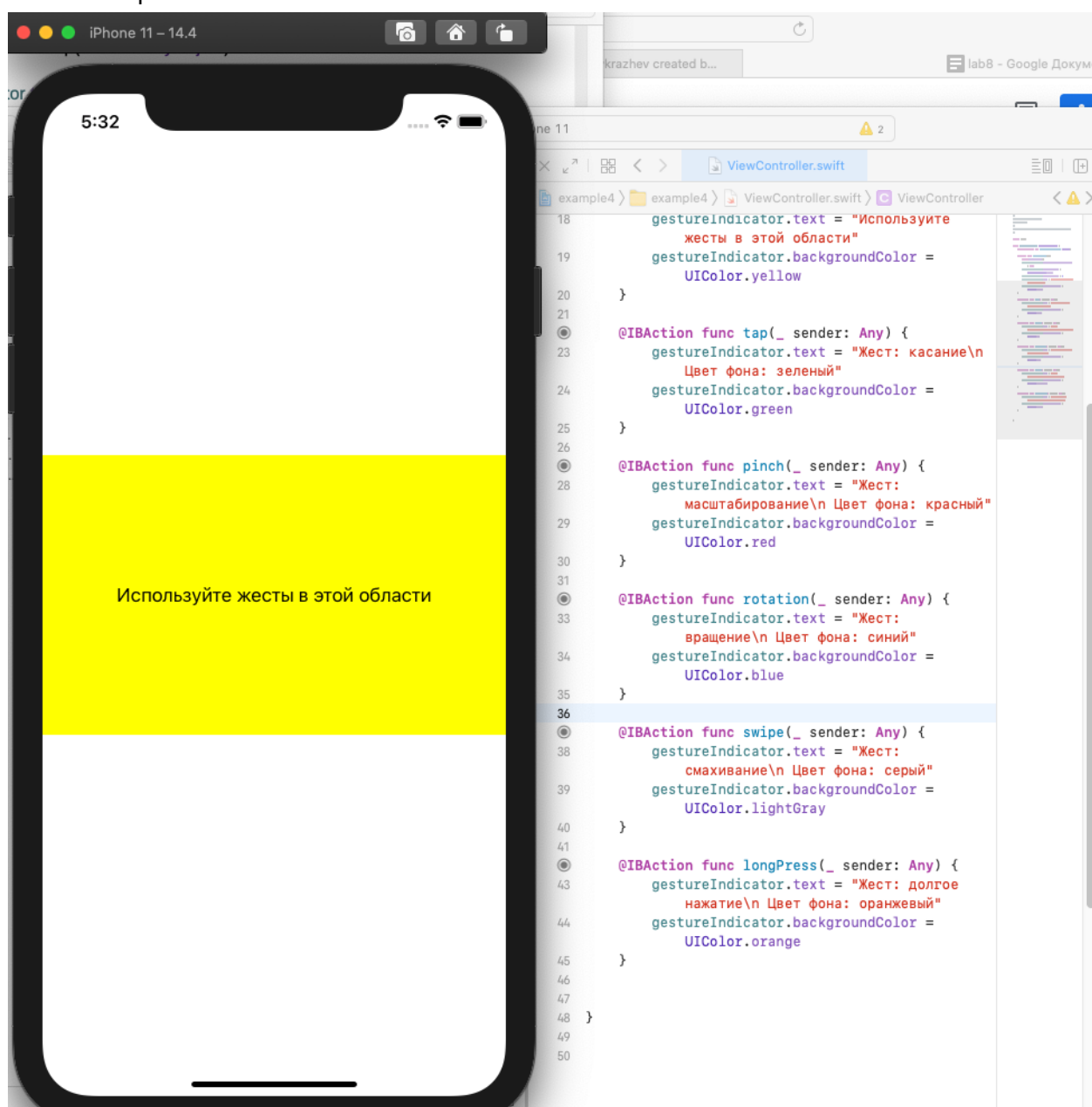
Цель работы: Научиться обрабатывать события, которые будут происходить при использовании различных жестов.

Создать объект UILabel, который будет отображать используемый пользователем жест и менять цвет своего фона, в зависимости от используемого жеста. Реализовать работу следующих жестов:

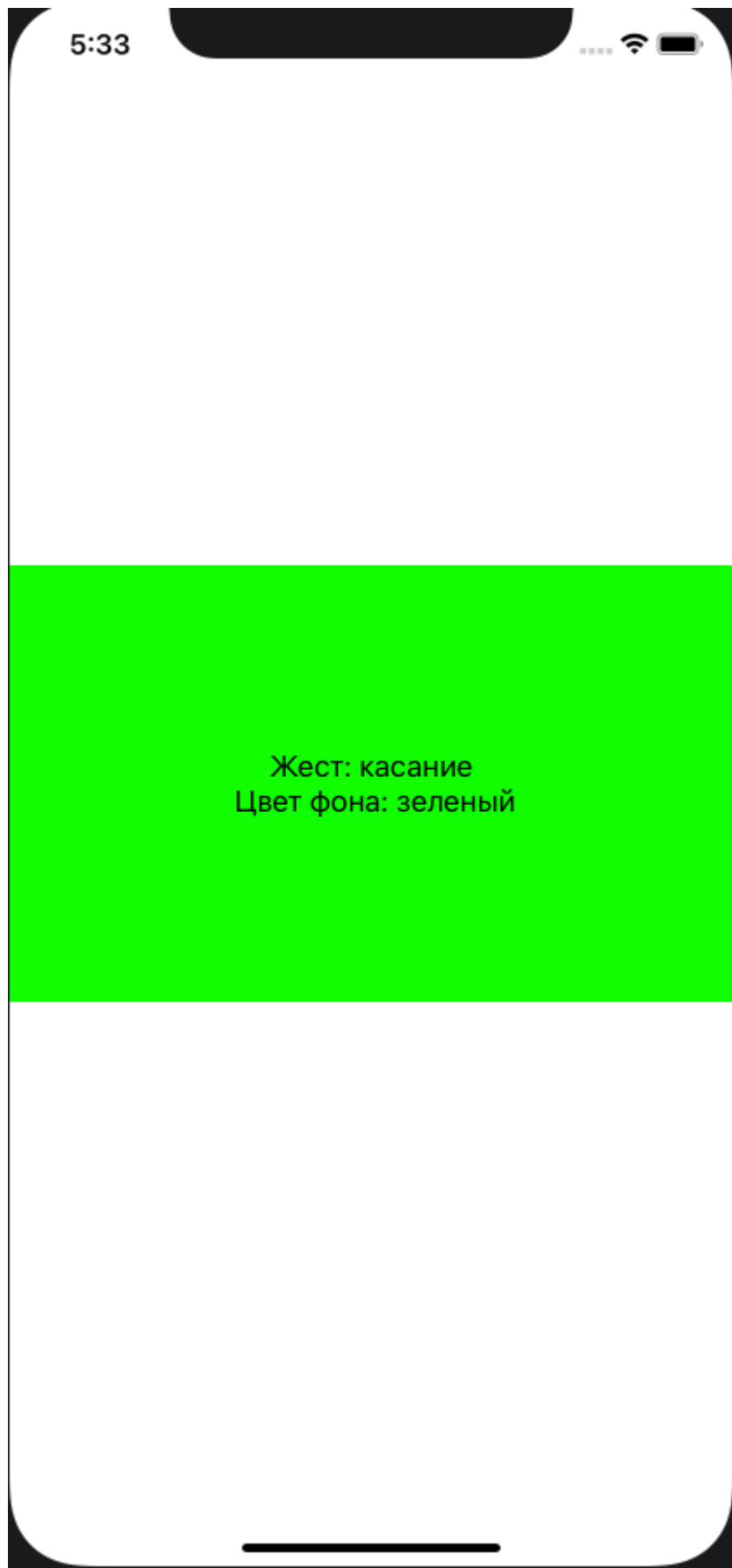
1. Вращение(цветфона:синий)
2. Масштабирование(цветфона:красный)
3. Касание(цветфона:зеленый)
4. Долгоенажатие(цветфона:оранжевый)
5. Смахивание(цветфона:серый)

Результат выполнения программы:

Главный экран:



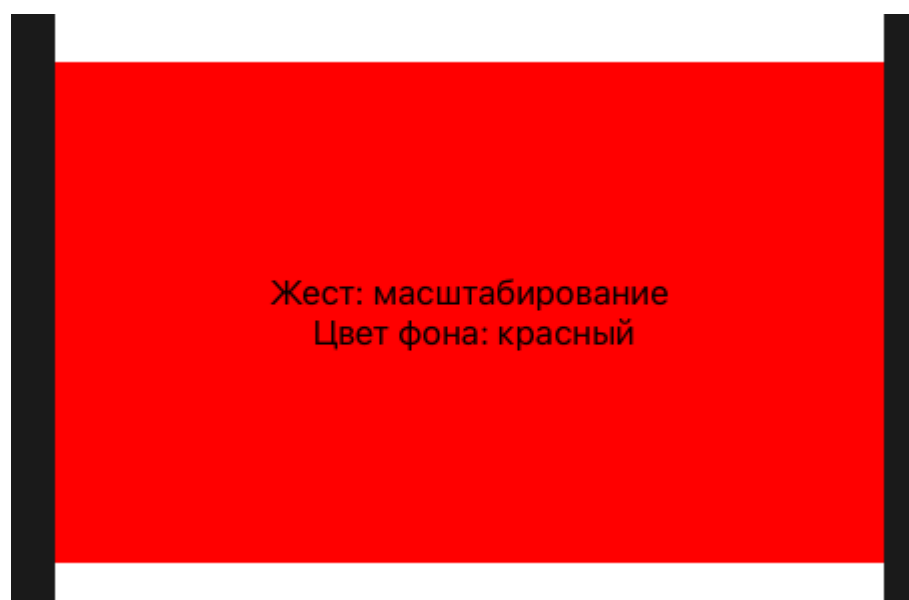
Таб:



Смахивание:



Pinch:



2. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

2.1 Рисование (приложение на языке objective-C)

Дополнить функционал приложения, созданного в 1.1. Требуется разработать программу рисования. Интерфейс программы должен поддерживать рисование линиями 5 размеров и разными цветами.

Реализовать возможность сохранения нарисованного изображения в файл.

Исходный код приложений на языке Objective-C должен соответствовать модели КИС для Objective-C.

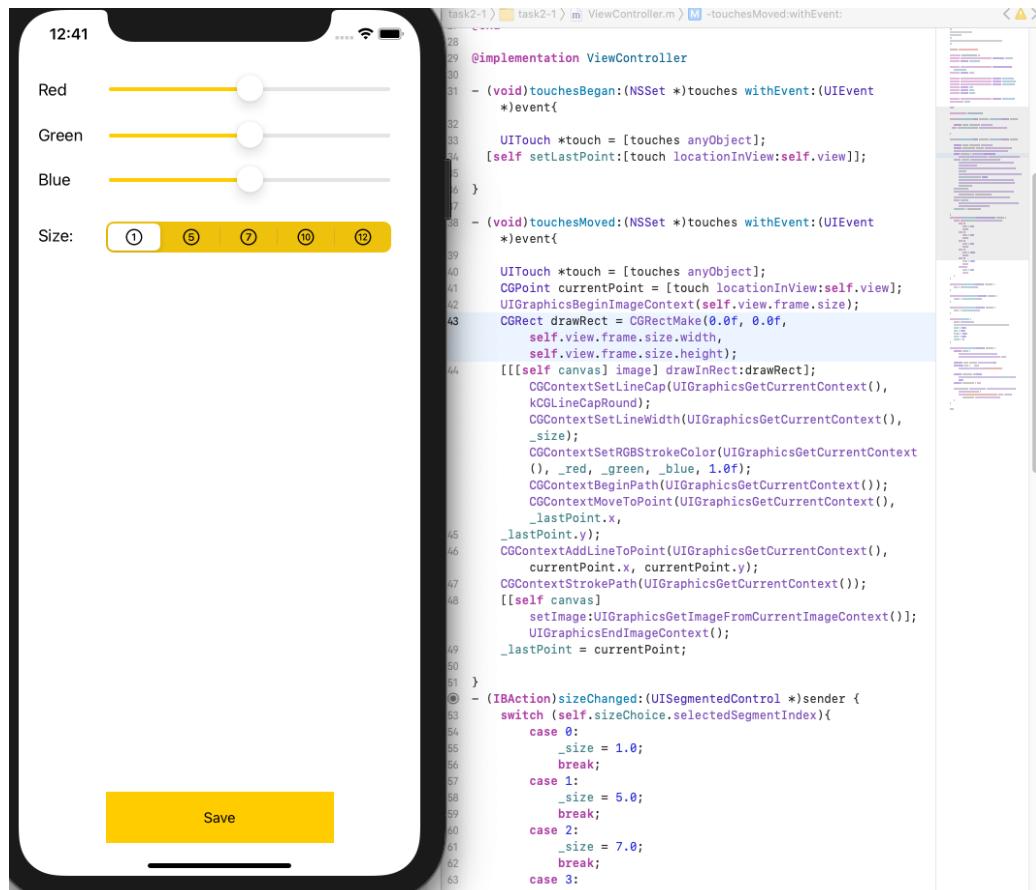
Входные данные

Цвет, размер и форма кисти, использующейся для рисования.

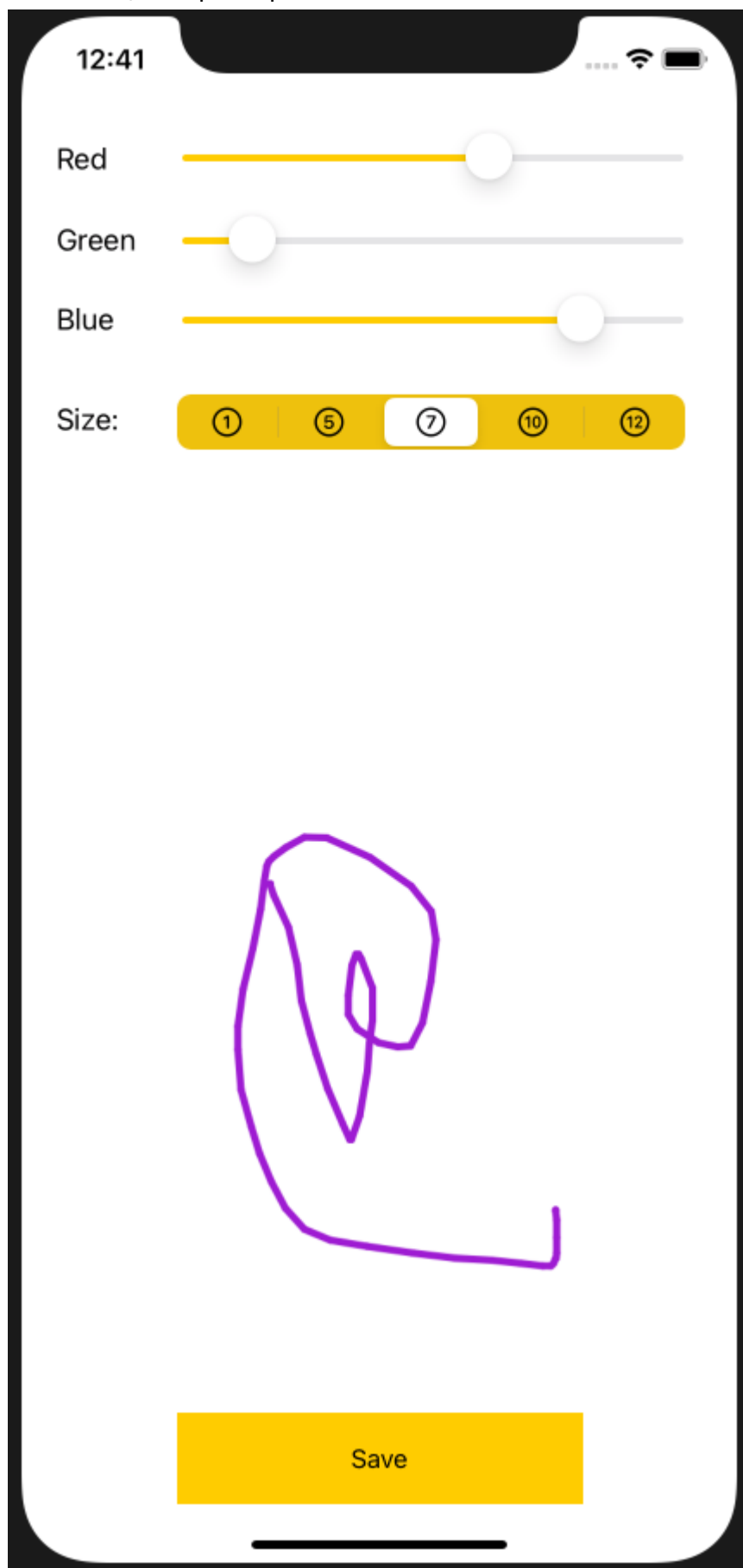
Выходные данные

Рисунок, состоящий из линий произвольной формы, сохранённый в библиотеку.

Главный экран полученного приложения:



Изменим цвет и размер кисти:



Еще раз:



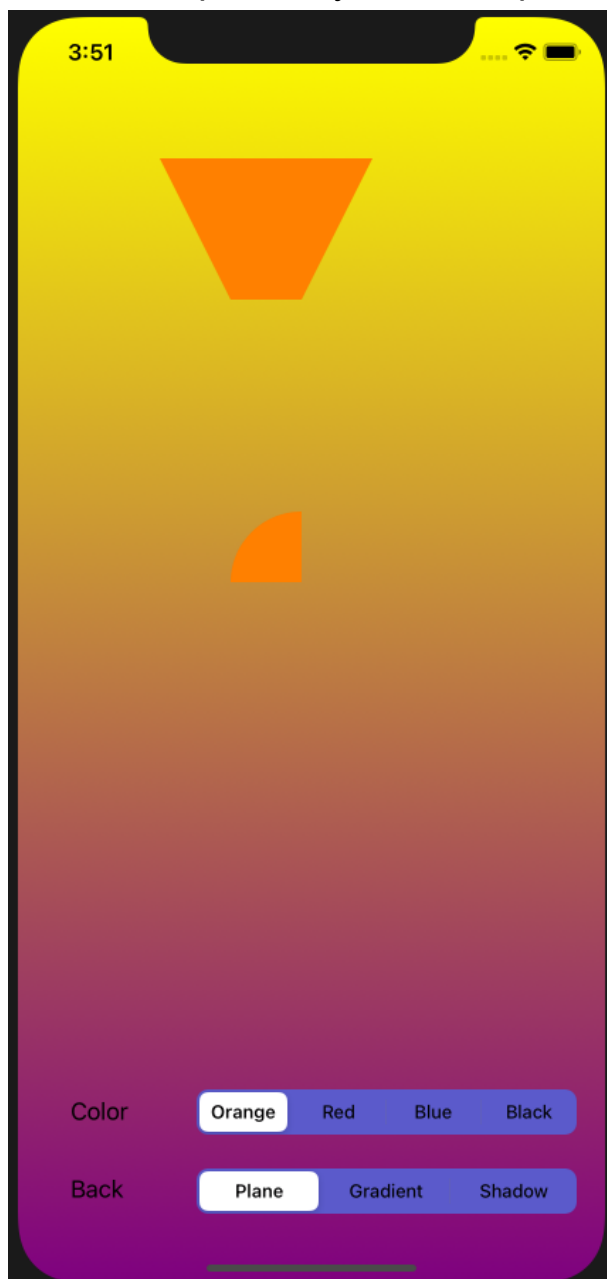
2.2. Создание iOS приложения с простыми графическими объектами на языке Swift

Создать iOS приложение, иллюстрирующее объекты двумерной графики с однотонным фоном, градиентом и тенью, согласно вариантам. Проиллюстрировать различные варианты создания новых фигур за счет вычитания или объединения 2 разных фигур.

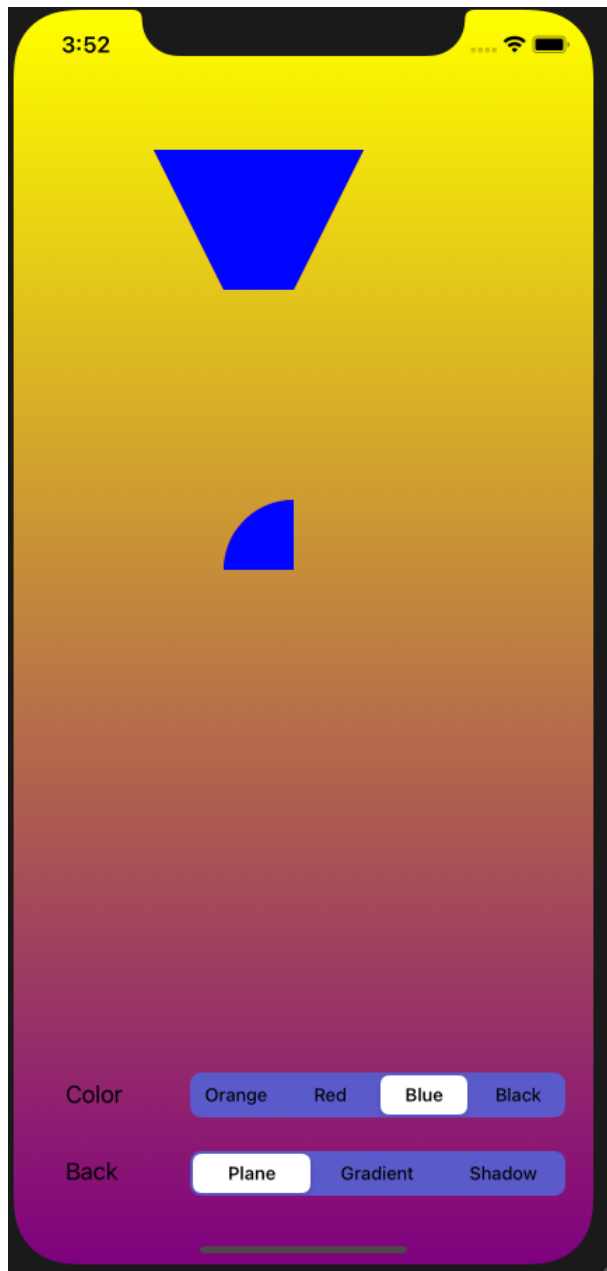
Код проекта распределить по группам (каталогам) согласно концепции MVC (Model, View, Controller).

Вариант: 34.Равнобедренная трапеция и круговой сектор.

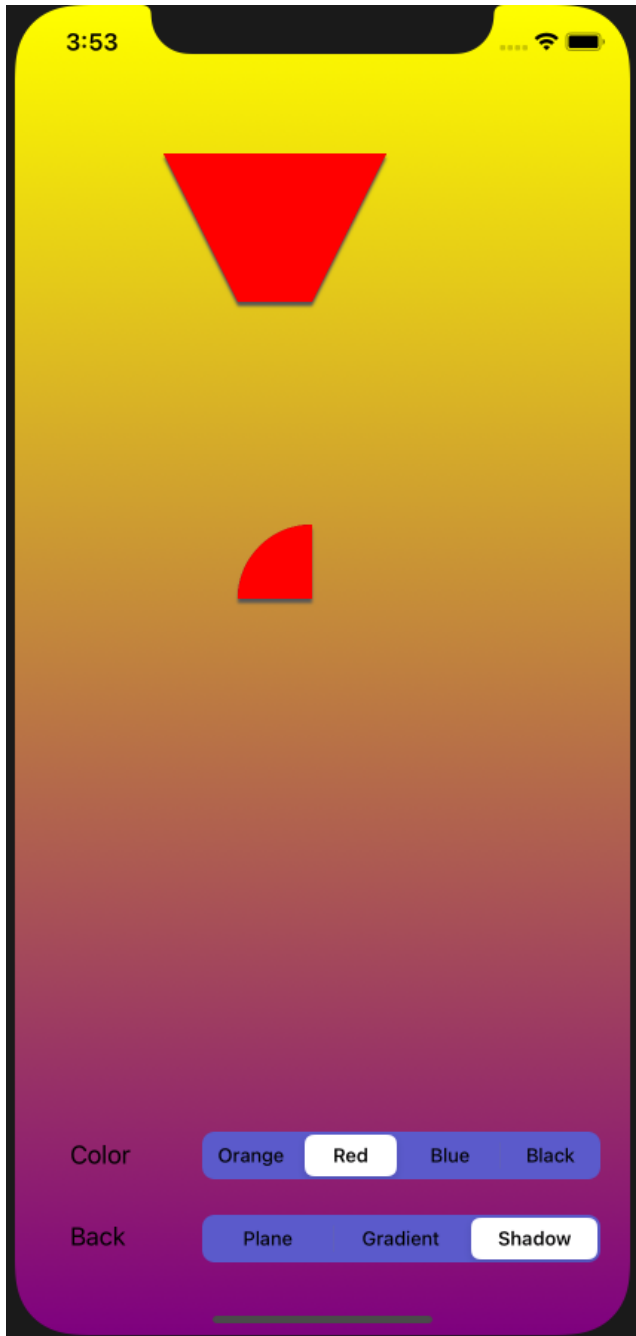
Главный экран полученного приложения:



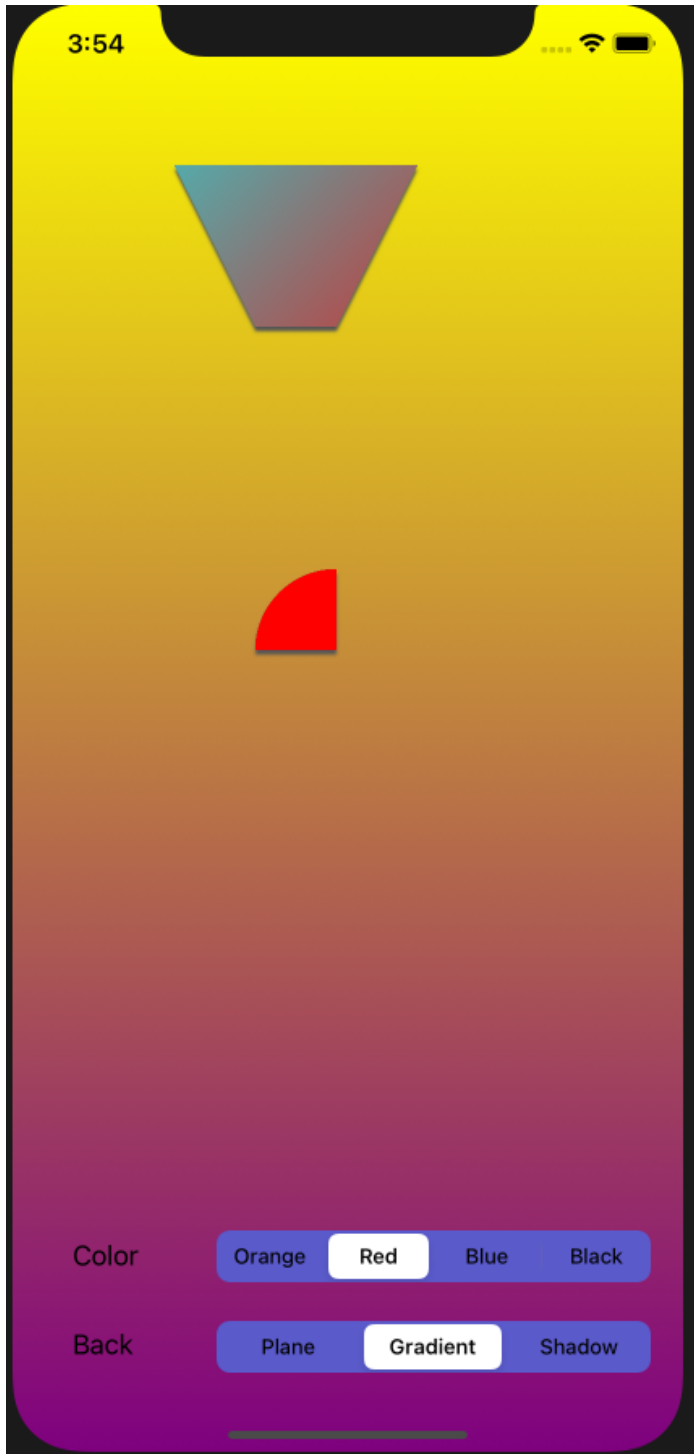
Попробуем поменять цвет:



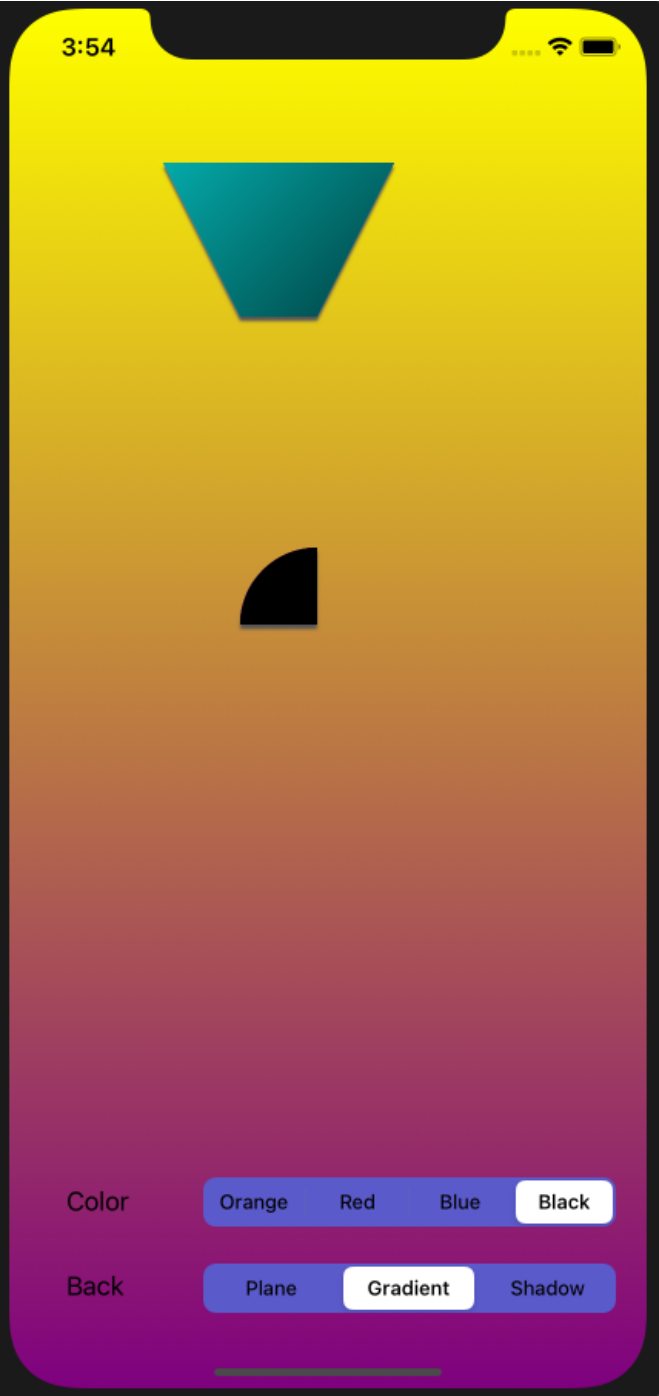
Включим тени:



Включим градиент для фигуры трапеция:



Поменяем цвет градиента:



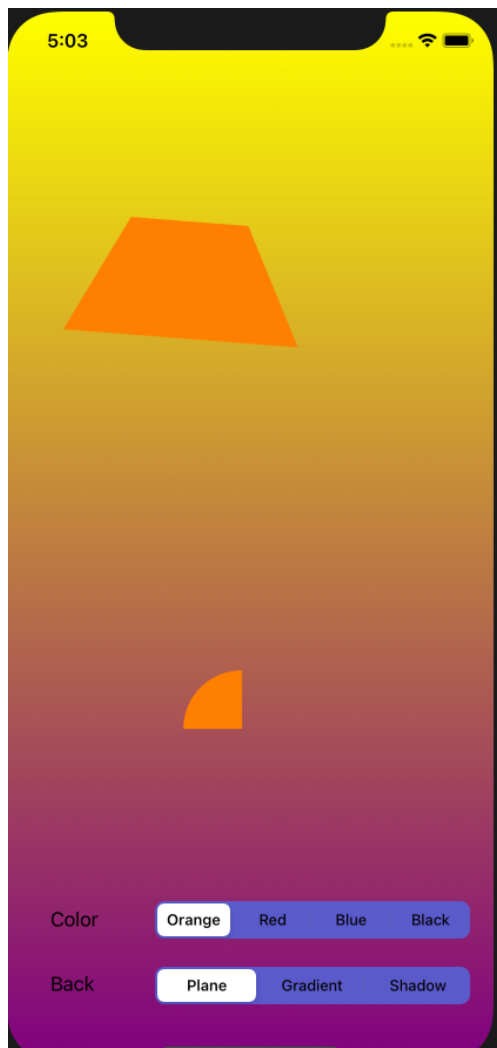
2.3. Создание iOS приложения с использованием анимации

Дополнить iOS приложение, созданное в задании 2.2, анимационными эффектами. Проиллюстрировать следующие эффекты:

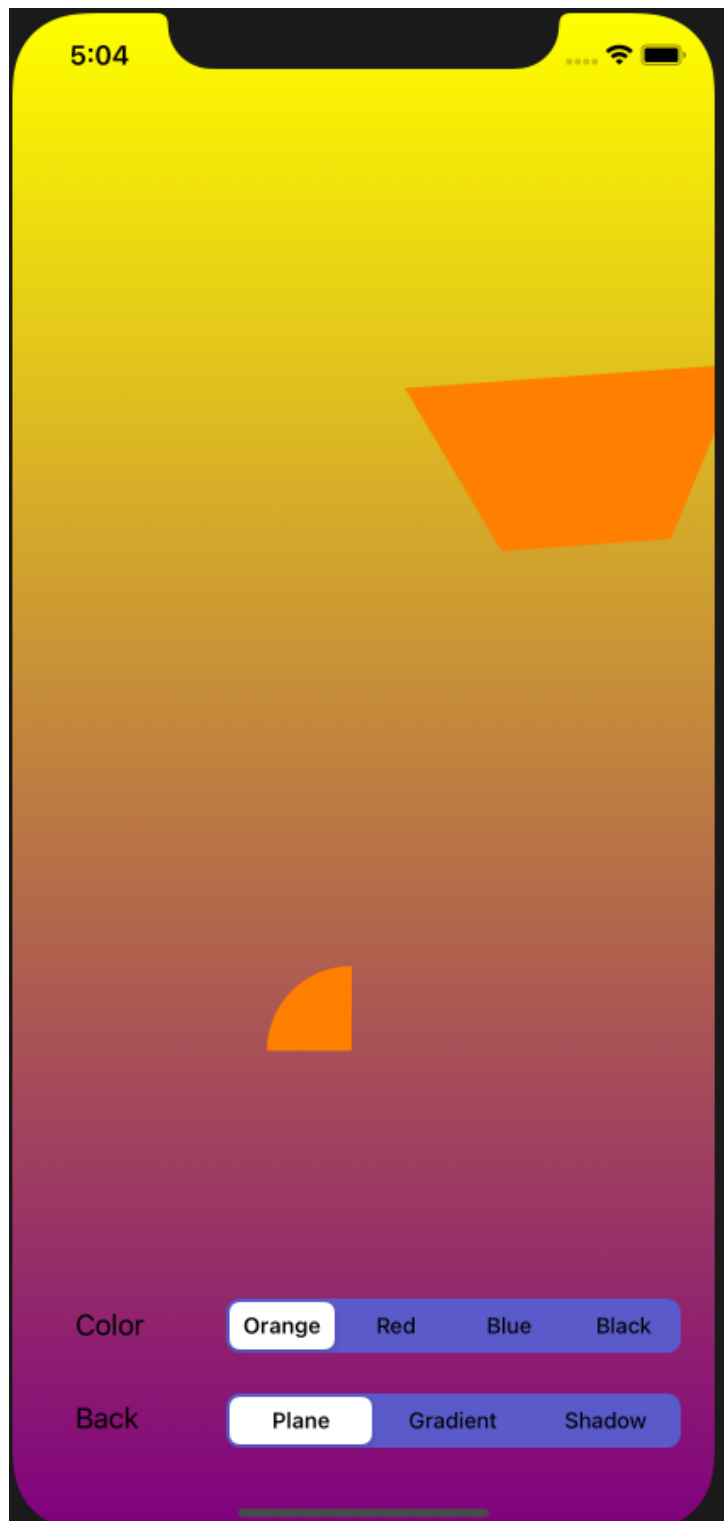
- Перемещение
 - Вращение
 - Масштабирование
 - Прозрачность
 - Наложение как минимум двух эффектов, например перемещение и вращение
- Код проекта распределить по группам (каталогам) согласно концепции MVC (Model, View, Controller).

В процессе выполнения была добавлена анимация появления двух segment control, анимация движения двух фигур, анимация вращения трапеции, анимация масштабирования сектора круга и анимация изменения прозрачности сектора при запуске приложения.

Главный экран:



В отчете показать анимацию довольно проблематично, но с ней можно ознакомиться при запуске проекта.



Как видно, трапеция летает(вращается) по всему экрану, а у сектора изменяется масштаб.

2.4. Создание iOS приложения на языке Swift с использованием жестов

Изменять фон одной из геометрических фигур, созданных в приложении в задании 2.2, в зависимости от жеста.

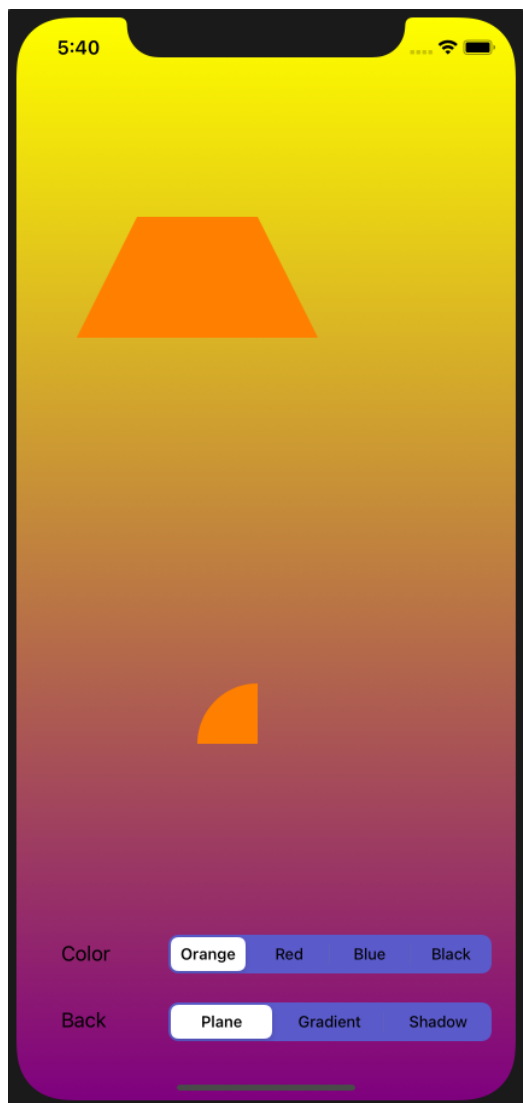
Проиллюстрировать следующие жесты:

1. Вращение (у фигуры устанавливается фон 1)
2. Масштабирование (у фигуры устанавливается фон 2)
3. Касание (у фигуры устанавливается фон 3)
4. Долгое нажатие (у фигуры устанавливается фон 4)
5. Смахивание (у фигуры устанавливается фон 5)

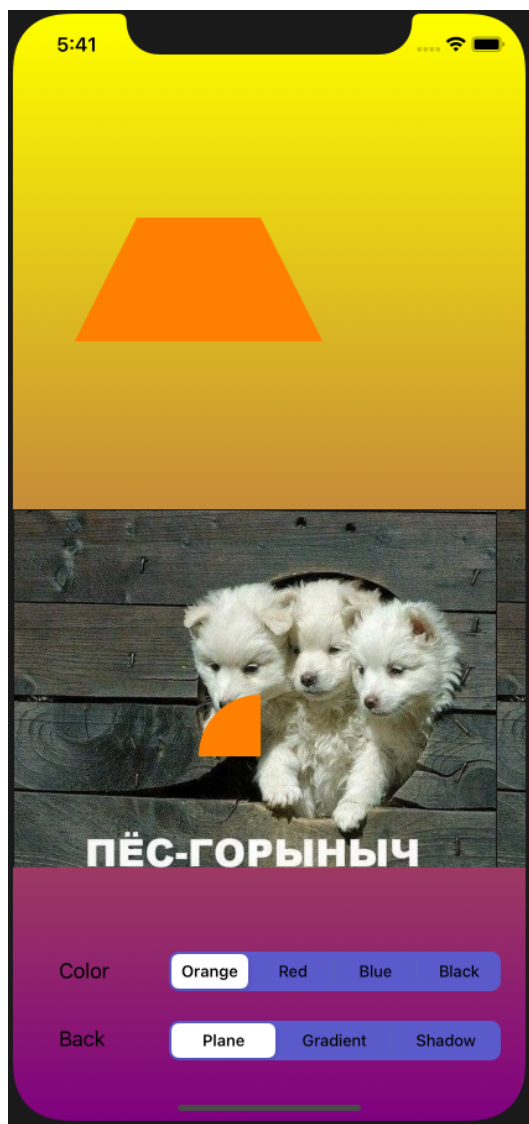
Для выполнения задания подобрать 5 изображений фона. Код проекта распределить по группам (каталогам) согласно концепции MVC (Model, View, Controller).

Результат выполнения работы

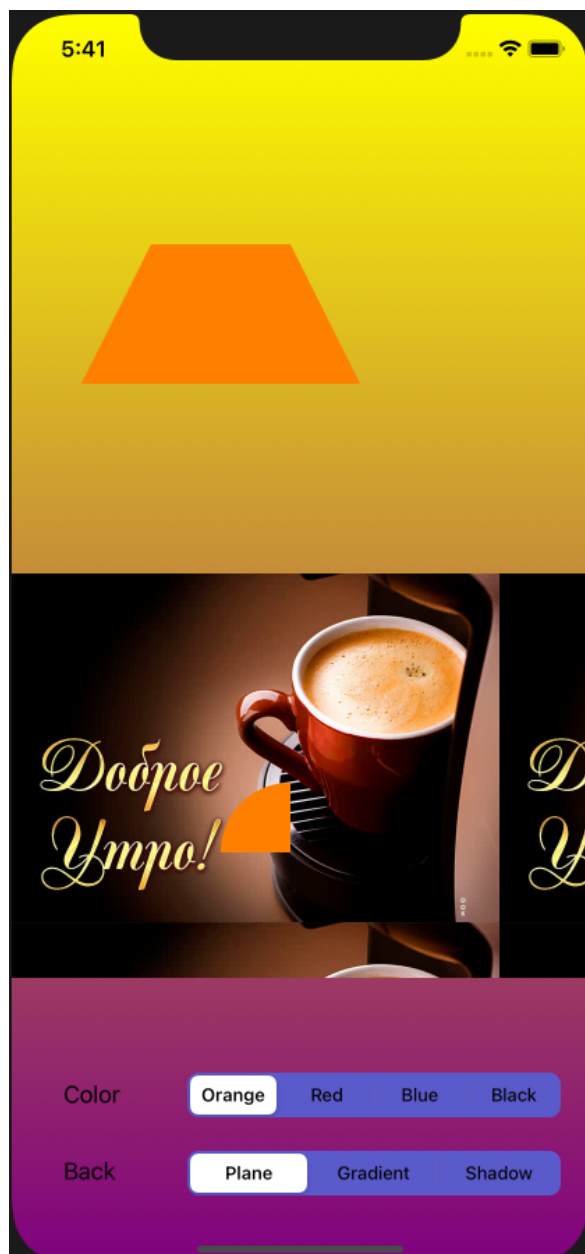
Главный экран:



Тап:



Долгое нажатие:



Свайп:



Масштабирование:



3. ЗАДАНИЕ ПОВЫШЕННОГО УРОВНЯ

3.1 Рисование 3D-графики на языке Swift

Разработать простую трехмерную игру, используя технологию OpenGL ES и язык программирования GLSL или Metal.

Трехмерная сцена должна поддерживать вращение и масштабирование объекта произвольной формы, например чайника/вазы или другого объекта, с помощью жестов.

Код проекта распределить по группам (каталогам) согласно концепции MVC (Model, View, Controller).

Входные данные

Массив вершин, хранящий позицию и цвет каждой вершины модели чайника/вазы или другого объекта.

Выходные данные

Изображение трёхмерной модели чайника/ кувшина / вазы / кольца.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1) Как нарисовать контур?

Сперва использовать команду `CGContextAddLineToPoint()`, а затем `StrokePath()`;

2) Как нарисовать прямоугольник?

Можно использовать следующие команды: `CGContext.addRect()`, затем `CgContext.drawPath()`

3) Как заполнить контур цветом? Градиентом?

Для этого используются команды `CGContext.FillPath()`, `DrawLinearGradient()`

4) Как заполнить прямоугольник цветом, сохранив цвет контура?

`Fill()` -> `Stroke()`;

5) Как нарисовать круг / эллипс?

```
var ovalPath = UIBezierPath(ovalInRect: CGRectMake(x, y, width, height))
UIColor.grayColor().setFill()
ovalPath.fill()
```

6) Как добавить тень к рисункам?

Следующей командой:

```
CGContextSetShadowWithColor(context, shadowOffset, shadowBlurRadius,
(shadow as UIColor).CGColor)
```

7) Как нарисовать изображение на контроллере View в прямоугольнике?

`Draw(in: rect)`

8) Как перерисовать представление View с помощью метода `SetNeedsDisplay`

Этот метод отправляет запрос на перерисовку при наступлении следующего цикла рисования.

9) Как вычесть одну фигуру из другой?

Нужно использовать `ImageView.mask`

10) Какой компонент Cocoa отвечает за обработку жестов?

`Gesture recognizer`

11) Как добавить и обработать жест касания (нажатия) (`tap gesture`)?

На нужный элемент перетащить `tap gesture recognizer`. Затем сделать действие для данного события.

12) Как добавить и обработать долгое нажатие (`long press gesture`)?

На нужный элемент перетащить `long press gesture recognizer`. Затем сделать действие для данного события.

13) Как добавить и обработать жест перелистывания (смахивания) (`swipe gesture`)?

На нужный элемент перетащить `swipe gesture recognizer`. Затем сделать действие для данного события.

14) Как обрабатывается жест стягивания (щипка) (`pinch gesture`)?

По аналогии с предыдущим.

15) Как обрабатывается жест растягивания (`spread gesture`)?

По аналогии с предыдущим.

16) Как добавить жесты непосредственно на Storyboard?

Открыть object library и перетащить необходимый элемент на storyboard.

17) Какие примитивы рисования Вы знаете?

Графические примитивы - минимальные графические объекты, которые составляют векторный рисунок.

Геометрические примитивы

18) Что такое CGContext?

Он нужен для хранения необходимой информации о рисовании (в т.ч. параметры рисования).

19) Что такое UIBezierPath? Чем отличается от CGContext?

20) Что такое CGImage?

Растровое изображение.

21) Что такое CGPath?

Описывает линии (или же фигуры) которые будут впоследствии нарисованы.

22) В чем заключаются трудности использования двумерной графики?

Сложно рисовать кривые - необходима высокая вычислительная характеристика компьютера.

23) Что такое OpenGL?

Это API(т.е. взаимодействует с графическим процессором) для рендеринга 2D и 3D векторной графики.

24) Что такое шейдер?

Программа, которая обрабатывает входной набор пикселей и возвращает новый. Выполняется на видеокарте.

25) Чем вершинный шейдер отличается от пиксельного?

Вершинные шейдеры позволяют выполнять математические действия над вершинами объектов, а пиксельные работают с пикселями (производят выборку из текстур, мат операции над цветом или глубиной) и выполняются попиксельно.

26) Что такое Metal?

Фреймворк для 3D графики (в основном), обеспечивающий доступ к графическому процессору.